

CLAIMS

1. A computer graphics evaluation and control system comprising:
 - a computer graphics system;
 - a graphics application configured to execute on and control said computer graphics system;
 - a graphics tool for evaluating and controlling said computer graphics system; and
 - a hooks module integrated within said computer graphics system for dynamically attaching said graphics tool to said computer graphics system during normal operations of said graphics application.
2. The system of claim 1,
 - wherein said computer graphics system comprises:
 - graphics hardware;
 - a graphics library for controlling said graphics hardware in accordance with function calls generated by said graphics application, including, graphics library functions responsive to said function calls and accessible through an applications program interface (API) defined by said function calls; and
 - wherein said hooks module comprises,
 - one or more API event generators each configured to provide said attached graphics tool access to operations related to an associated one of said graphics library functions in response to event requests received from

13 said graphics tool, and to provide said graphics tool with results of said
14 access, and

15 means for installing said one or more API event generators into said
16 graphics library.

1 3. The system of claim 2, wherein said one or more API event generators
2 comprises:

3 means for enabling said attached graphics tool to monitor said selected
4 operations related to said associated graphics library function.

5
6 4. The system of claim 2, wherein said one or more API event generators
7 comprises:

8 means for enabling said attached graphics tool to control said selected
9 operations related to said associated graphics library function.

1 5. The system of claim 2,
2 wherein said computer graphics system further comprises,
3 control modules for controlling said graphics hardware in response to
4 said function calls received through said API; and
5 wherein said hooks module further comprises,
6 one or more internal event generators each configured to provide said
7 attached graphics tool access to an associated function in said control
8 modules, and

9 means for enabling said internal event generators in said graphics
10 library.

1 6. The system of claim 5, wherein said installing means comprises:
2 means for dynamically redirecting one or more of said function calls
3 generated by said graphics application to said API event generators, wherein said
4 API event generators include means for calling said graphics library functions
5 identified in said redirected function call.

1 7. The system of claim 6, wherein said redirecting means comprises:
2 a normal operations dispatch table including function pointers to said
3 graphics library functions;
4 a hooks dispatch table including function pointers to said API event
5 generators; and
6 means for copying said normal operations dispatch table and said hooks
7 dispatch table to an active dispatch table in said graphics library.

1 8. The system of claim 5, wherein said graphics library includes pipeline
2 control modules for managing a graphics pipeline in said graphics system and
3 device-specific modules for controlling components of said graphics hardware,
4 wherein said internal event generators comprise:
5 pipeline control module event generators for providing said attached
6 graphics tool access to said pipeline control modules; and

7 device-specific module event generators for providing said attached
8 graphics tool access to said device-specific control modules.

1 9. The system of claim 2, wherein said graphics tools and said hooks module
2 communicate with each other through a graphics tool interface.

1 10. The system of claim 9, wherein said graphics tool interface is a interprocess
2 communications (IPC) mechanisms providing socket communications between
3 said graphics tools and said hooks module.

1 11. A hooks system for providing a graphics tool access to a computer graphics
2 system to evaluate and control a graphics application executing on the computer
3 graphics system, the system comprising:

4 application program interface (API) event generators for performing
5 predetermined operations relating to a graphics library function call and for
6 generating a hook event including results of said predetermined operations;

7 dispatch table manger for selecting an active dispatch table from a normal
8 operations dispatch table having function pointer to said graphics library functions
9 and a hooks dispatch table having pointer to said API event generators; and

10 a hook event manager for enabling and configuring selected ones of said
11 API event generators in response to a graphics tool event request.

1 12. The system of claim 10, further comprising:

2 internal event generators, integrated along various locations of a graphics
3 pipeline managed by said graphics library, for performing predetermined
4 diagnostic operations in said graphics system;

5 wherein said hook event manager also enables and configures selected ones
6 of said internal event generators.

1 13. The system of claim 12, wherein said one or more API event generators
2 comprises:

3 means for enabling said attached graphics tool to monitor said selected
4 operations related to said associated graphics library function.

5
6 14. The system of claim 12, wherein said one or more API event generators
7 comprises:

8 means for enabling said attached graphics tool to control said selected
9 operations related to said associated graphics library function.

1 15. The system of claim 111, wherein said dispatch table manager comprises:

2 means for dynamically redirecting one or more of said function calls
3 generated by said graphics application to said API event generators, wherein said
4 API event generators include means for calling said graphics library functions
5 identified in said redirected function call.

1 16. The system of claim 15, wherein said redirecting means comprises:

2 a normal operations dispatch table including function pointers to said
3 graphics library functions;

4 a hooks dispatch table including function pointers to said API event
5 generators; and

6 means for copying either said normal operations dispatch table or said
7 hooks dispatch table to an active dispatch table in said graphics library.

1 17. The system of claim 15, wherein said graphics library includes pipeline
2 control modules for managing a graphics pipeline in said graphics system and
3 device-specific modules for controlling components of said graphics hardware,
4 wherein said internal event generators comprise:

5 pipeline control module event generators for providing said attached
6 graphics tool access to said pipeline control modules.

1 17. The system of claim 15, wherein said graphics library includes pipeline
2 control modules for managing a graphics pipeline in said graphics system and
3 device-specific modules for controlling components of said graphics hardware,
4 wherein said internal event generators comprise:

5 pipeline control module event generators for providing said attached
6 graphics tool access to said pipeline control modules.

1 18. The system of claim 17, wherein said graphics library includes pipeline
2 control modules for managing a graphics pipeline in said graphics system and

device-specific modules for controlling components of said graphics hardware,
wherein said internal event generators further comprises:
device-specific module event generators for providing said attached
graphics tool access to said device-specific control modules.

19. The system of claim 18, wherein said device-specific module event
generators further provide said attached graphics tool access to components of
said graphics hardware.

20. A method for proving a graphics tool asynchronous access to a computer
graphics system, comprising the steps of:

(a) installing an API event generator in a graphics library of the graphics
system, said API event generator associated with a graphics library function;

(b) receiving a function call issued from a graphics application to invoke
a graphics application function;

(c) routing said function call to said API event generator;

(d) performing selected operations related to said associated graphics
library function;

(e) calling the associated graphics library function when said associated
graphics library function is to be called;

(f) forwarding results of said selected operations to said graphics tool;

and

(g) logging timing information with said graphics tool.